REMARKS

Applicants respectfully request the Examiner's reconsideration of the present application.

Claims 1, 3, and 5-22 in the present application are pending.

Claims 1, 3, 5-7, 9, and 11-22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,711,602 ("Bhandal") in view of U.S. Patent No. 7,046,723 ("Schier").

Claims 8 and 10 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claims 1, 3, 5-7, 9, and 11-22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over 6,711,602 ("Bhandal") in view of U.S. Patent 7,046,723 ("Schier").

It is submitted that Bhandal and Schier do not render claims 1, 3, 5-7, 9, and 11-22 unpatentable under 35 U.S.C. §103(a).

Bhandal includes a disclosure of a pair of parallel 16.times.16 multipliers each with two 32-bit inputs and one 32-bit output. There are options to allow input halfword and byte selection for four independent 8x8 or two independent 16x16 multiplications, real and imaginary parts of complex multiplication, pairs of partial sums for 32x32 multiplication, and partial sums for 16x32 multiplication. There are options to allow internal hardwired routing of each multiplier unit results to achieve partial-sum shifting as required to support above options. There is a redundant digit arithmetic adder before final outputs to support additions for partial sum accumulation, complex multiplication vector accumulation and general accumulation for FIRs/IIRs--giving MAC unit functionality. There are options controlled using bit fields in a control register passed to the multiplier unit as an operand. There are also options to generate all of the products needed for complex multiplication (see Bhandal Abstract).

Schier includes a disclosure of a digital and a multiplication method are described, which lead to an efficient architecture for a hardware implementation of digital FIR and IIR filters into

FPGAs. The multiplications of input sample data and delayed sample data with filter coefficients are performed by addressing look-up tables in which corresponding multiplication results are prestored. The size of the look-up tables is reduced by storing only those multiplication results which cannot be obtained by a shifting operation performed on the other pre-stored multiplication results, the input sample data, or the delayed sample data. Thereby, the size of the look-up tables can be compressed significantly such that an implementation of large digital filters into FPGAs is possible (see Schier Abstract).

It is submitted that Bhandal and Schier do not teach or suggest a method for performing multiplication of a first number with a second number on a field programmable gate array utilizing a single digital signal processor (DSP) configured to multiply only a subset of a number of bits forming the first and second numbers that includes generating a product by multiplying a first plurality of bits from a first number and a first plurality of bits from a second number using the DSP configured to perform multiplication on a fewer number of bits then those forming the first and second numbers, retrieving a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory, scaling the product with respect to a position of the first plurality of bits from the first number and a position of the first plurality of bits from the second number and scaling the stored value with respect to a position of the second plurality of bits from the first number and a position of the second plurality of bits from the second number, and summing a scaled product and a scaled stored value to generate a value representing a product of the first number and the second number, wherein the first number and the second number each have a number of bits equal to or greater than a total of the first and second plurality of bits.

In the Office Action mailed 6/10/2009, the Office states in part that

> Re claim 1, Bhandal et al. disclose in Figures 1-22 a method for performing multiplication (e.g. abstract and Figure 8 as general architecture of multiplier), comprising: generating a product by multiplying a first plurality of bits from a first number and a second plurality of bits from a second number (e.g. Figure 11B wherein SRC1_L as B is multiplying with SRC2_L as D by B*D) using a digital signal processor (DSP) configured to

perform multiplication on a fewer number of bits then those forming the first and second numbers (e.g. by multiplier 800 in Figure 8 wherein the multiplier 800 performs 16 bits multiplication of 32 bits operands); a product of a second plurality of bits from the first number and a second plurality of bits from the second number (e.g. Figure 11 wherein SRC1_H as A is multiplying with SRC2_H as C by A*C); scaling the product with respect to a position of the first plurality of bits from the first number and a position of the first plurality of bits from the second number (e.g. by shifter 810 in Figure 8) and scaling the stored value with respect to a position of the second plurality of bits from the first number and a position of the second plurality of bits from the second number (e.g. by shifter 811 in Figure 8); and summing a scaled product and a sealed (sic) stored value to generate a value representing a product of the first number and the second number (e.g. output of adder 820 in Figure 8 and Figure 11B), wherein the first number and the second number each have a number of bits equal to or greater than a total of the first and second plurality of bits (e.g. Figure 11B wherein each of input operands consist of 32 bits and each of input multiplied operand is 16 bits).

Bhandal et al. fail to disclose the multiplier is on a field programmable gate array and that the second product is retrieved from a memory. However, Schier et al. disclose in Figures 1-4 the multiplier is on a field programmable gate array (e.g. abstract) and the second product is retrieved from a memory (e.g. any intermediate product from the LUT in Figures 1-4 as b|x).

(6/10/2009 Office Action, pp. 2-3) (Emphasis Added).

The Office acknowledges that Bhandal does not disclose "retrieving a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory" and "scaling the stored value with respect to a position of the second plurality of bits from the first number and a position of the second plurality of bits from the second number" as required by the present invention.

Applicants submit that the Office cannot properly combine the "intermediate product from the LUT in Figures 1-4" of Schier with Bhandal to be used as "the second product" which is scaled by shifter 811 in Figure 8 of Bhandal as the Office suggests on page 3 of the Office Action mailed 6/10/2009.

When combining two references, if the proposed modification would render the prior art unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the

proposed modification. In re Gordon, 733 F.2d 900 (Fed.Cir. 1984) (also see MPEP 2143.01 V). Schier clearly states in its Summary of the Invention, that one of the objects of the present invention is achieved by having its input sample data not divided into its bit position so that an additional shifting operation is not required after multiplication and a low latency is introduced (see Schier column 3, columns 3-11 and column 4, lines 3-12). Applicants submit that combining Schier and Bhandal in the manner suggested by the Office would run contrary to established case law since it would render the invention of Schier unsatisfactory for its intended purpose.

Furthermore, Applicants disagree with the Office's motivation for combining Bhandal with Schier when the Office states that "it would have been obvious to a person having ordinary skill in the art at the time of the invention is made to add ... the second product is (sic) retrieved from a memory as seen in Schier et al.'s invention into Bhandal et al.'s invention because it would enable to improve system performance." Applicants submit that the example in Bhandal discloses that the number of bits of the numbers being multiplied, 32, match the configuration of the number of bits multiplied by multipliers 800 and 801, 32. Thus, there is no need or motivation to add a further product from memory. Furthermore, because there is a match, there would be no improvement in system performance. In fact, by requiring a stored product to be added, the multiplier in Bhandal would have its performance worsen not improve.

In the Office Action mailed 6/10/2009, the Office states in part that

> The applicant argues in pages 12-13 for claims that the combination of reference is improper since the combination would worsen the combination rather than improving the invention and further there is no need or motivation to add a further product from the memory as required.
>
> The examiner respectfully submits that neither the references would explicitly state the incompatible of references and further states that the invention would worsen if the two is combined. Generally, the missing limitation from primary reference is only getting the second product from the memory for yielding complete four 8x8 multiplications. To yield proper result of 32x32 multiplication with four 8x8 multiplication using only two multipliers, there must be the four multiplications for four 8x8 multiplications. Thus, the second product is the

> previous result of pair multiplications to achieve 32 bits multiplication result, see Figure 9 with operation 0001 as four 8x8 multiplication to achieve 32 bit multiplications.

(6/10/2009 Office Action, pp. 13-14).

Applicants submit that Bhandal clearly supports performing four independent 8x8 multiplications without requiring Bhandal to perform "getting the second product from the memory for yielding the complete four 8x8 multiplications". For example, Bhandal discloses the following.

> In total there are nine 32-bit adders, four 32-bit shifters, three Boolean operators, and two 32x16 multipliers. The multipliers are each configurable into two 16x16 or four 8x8 multipliers.

(Bhandal column 5, lines 11-15)

> In general, M multiply unit 171 can perform the following operations: two 16x16 multiplies or four 8x8 multiplies with all combination of signed or unsigned numbers, Q-shifting and A-shifting of multiply results, rounding for extended multiply (EMPY) instructions, controlling the carry chain by breaking/joining the carry chain at 16-bit block boundaries, and saturation multiplication where the final result is shifted left by 1 or returns 0x7FFFFFFF if an overflow occurs. Multiplication is broken down into three stages, starting with Multiply Parts IA & IB 173, which provide the inputs for Multiply Parts IIA & B 175, followed by the final stage which contains Adder/Converter 177 and quotient shifter (Q-shift) 179.

(Bhandal column 6, lines 15-27).

> Advantageously, paired multipliers 800, 801 can be configured to perform: two 16x16 multiply or four 8x8 multiply with all combination of signed/unsigned numbers, Q-shifting and A-shifting of multiply results, support rounding for EMPY instructions, control the carry chain by breaking/joining the carry chain at 16-bit block boundary, and support saturation multiply where the final result is shifted left by 1 or return 0x7FFHTFF if overflow occurred.

(Bhandal column 8, lines 3-10)

> FIG. 11G illustrates a quad 8-bit by 8-bit multiplication that produces A*E, B*F, C*G, and D*H. This produces four independent products for two operands with packed data using the PEMPHL instruction with the EMPY register set-up to configure the multiply units as four individual 8x8 bit multipliers.

(Bhandal column 8, line 66 through column 9, line 4).

Thus, Applicants submit that the combination of Bhandal with Schier lacks motivation and is improper by rendering the prior art unsatisfactory for its intended purpose.

In contrast, claim 1 states

> A method for performing multiplication of a first number with a second number on a field programmable gate array utilizing a single digital signal processor (DSP) configured to multiply only a subset of a number of bits forming the first and second numbers, comprising:
> generating a product by multiplying a first plurality of bits from the first number and a first plurality of bits from the second number using the DSP configured to perform multiplication on a fewer number of bits then those forming the first and second numbers;
> retrieving a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory;
> scaling the product with respect to a position of the first plurality of bits from the first number and a position of the first plurality of bits from the second number and scaling the stored value with respect to a position of the second plurality of bits from the first number and a position of the second plurality of bits from the second number; and
> summing a scaled product and a scaled stored value to generate a value representing a product of the first number and the second number, wherein the first number and the second number each have a number of bits equal to or greater than a total of the first and second plurality of bits.

(Claim 1, as Amended) (Emphasis Added).

Claims 11, 17, 21, and 22 include similar limitations.

Given that claims 3, and 5-10 depend from claim 1, claims 12-16 depend from claim 11, and claims 18-20 depend from claim 17, it is likewise submitted that claims 3, 5-10, 12-16, and 18-20 are also patentable under 35 U.S.C. §103(a) over Bhandal and Schier.

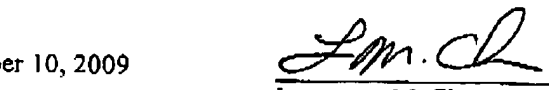
In view of the arguments set forth herein, it is respectfully submitted that the applicable rejections and have been overcome. Accordingly, it is respectfully submitted that claims 1, 3, and 5-22 should be found to be in condition for allowance.

The Examiner is invited to telephone Applicants' attorney (217-377-2500) to facilitate prosecution of this application.

If any additional fee is required, please charge Deposit Account No. 50-1624.

Respectfully submitted,

Dated: September 10, 2009



Lawrence M. Cho
Attorney for Applicants
Registration No. 39,942

P.O. Box 2144
Champaign, IL 61825
(217) 377-2500